

[Sign in](#)[Google](#)[Web](#) [Images](#) [Video](#) [News](#) [Maps](#) [more »](#)

verification Simulink code

[Search](#)[Advanced Search](#)
[Preferences](#)**Web**Results 1 - 10 of about 217,000 for **verification Simulink code**. (0.35 seconds)**The MathWorks - Seminar- ECU Production Code Generation and ...**

Learn how **Simulink** enables common software engineering tasks such as software ...
ECU Production **Code Generation** and **Verification** Using **Simulink** ...
www.mathworks.com/company/events/seminars/seminar6919.html?by=date - 18k -
[Cached](#) - [Similar pages](#)

The MathWorks - Recorded Webinar: Embedded Code Generation ...

It then shows how the models and generated **code** can be tested and analyzed using
Simulink Verification and Validation along with in-the-loop testing. ...
www.mathworks.com/wbmr8590 - 47k - [Cached](#) - [Similar pages](#)
[[More results from www.mathworks.com](#)]

Research Results | Documenting Failures of MATLAB/Simulink Code ...

Objective, By 04/30/2005 we expect to outline the procedure to perform the
verification of **Simulink** blocks converted into C **code** using RTW. ...
sarpresults.ivv.nasa.gov/ViewResearch/91.jsp - 8k - [Cached](#) - [Similar pages](#)

The MathWorks United Kingdom - Seminar- High-Integrity Code ...

learn how **Simulink** enables common software engineering tasks such as detailed
software ... High-Integrity **Code Generation** and **Verification** Using **Simulink** ...
www.mathworks.co.uk/company/events/seminars/seminar11518.html - 23k -
[Cached](#) - [Similar pages](#)

The MathWorks Schweiz - Simulink Code Generation Application Day ...

Model-Based Design with **Simulink**. Rapid Prototyping and Hardware-in-the-Loop for
testing and **verification** of Designs; High Integrity **Code Generation** and ...
www.mathworks.ch/company/events/seminars/fin_infoday.html - 22k -
[Cached](#) - [Similar pages](#)

The MathWorks Benelux - Press Room - The MathWorks Introduces ...

"Generating HDL **code** from **Simulink** models has been a top item on their wish ... to
write 10 lines of HDL **verification code** for every line of hardware **code**. ...
www.mathworks.nl/company/pressroom/articles/article13278.html - 26k -
[Cached](#) - [Similar pages](#)

[PDF] Saber-MATLAB Integrations: Enabling Virtual HW/SW Co-Verification

File Format: PDF/Adobe Acrobat - [View as HTML](#)
HW/SW co-**verification** is initiated early in the design process, where control ...
C-**code** for a **Simulink**/Stateflow model that is then compiled as a ...
www.synopsys.com/products/mixedsignal/saber/cosim_linkmatlab_wp.pdf -
[Similar pages](#)

The MathWorks España - Press Room - The MathWorks Introduces Link ...

Shown here is an example of system-level **verification** of a video edge detection
system in a **Simulink** test bench with DSP **code** executing on a TI C6713DSP and ...
www.mathworks.es/company/pressroom/articles/article12723.html - 26k -
[Cached](#) - [Similar pages](#)

[PDF] AUTOMATIC CO-VERIFICATION OF FPGA DESIGNS IN SIMULINK

File Format: PDF/Adobe Acrobat - [View as HTML](#)
Verification of DSP systems is an error-prone and time- consuming process,
because many ... The first branch "**Simulink code** generation and compilation" ...
www.nt.tu.wien.ac.at/rapid_prototyping/
publications/papers_rupp/MBD05Brandmayrc.pdf - [Similar pages](#)

Embedded.com - Simulink adds Verification and Validation tool

Simulink's Verification and Validation is an add-on tool to **Simulink** that enables ...
as comments in the embedded **code** generated from the validated model. ...
www.embedded.com/showArticle.jhtml?articleID=159402892 - 64k -
[Cached](#) - [Similar pages](#)


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

Verification of software code Simulink model



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **Verification of software code Simulink model**

Found 96,859 of 198,991

Sort results by [Save results to a Binder](#)[Try an Advanced Search](#)Display results [Search Tips](#)Try this search in [The ACM Guide](#)☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐**1** [Search-based software engineering: papers: The state problem for test generation in](#)[Simulink](#)

Yuan Zhan, John A. Clark

July 2006 **Proceedings of the 8th annual conference on Genetic and evolutionary computation GECCO '06**

Publisher: ACM Press

Full text available: [pdf\(417.23 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Search based test-data generation has proved successful for code-level testing. In this paper we investigate the application of such approaches at the higher levels of abstraction offered by *Matlab-Simulink* models. The presence of persistent state has been shown to be problematic at the code level and such difficulties remain when *Matlab-Simulink* models are to be tested. In such cases, sequences of inputs that can put the model under test into particular states are needed to enable ...

Keywords: Matlab-Simulink, automation, state problem, structural coverage, test-data generation, tracing and deducing

2 [Search-based software engineering: Search-based mutation testing for *Simulink* models](#)

Yuan Zhan, John A. Clark

June 2005 **Proceedings of the 2005 conference on Genetic and evolutionary computation GECCO '05**

Publisher: ACM Press

Full text available: [pdf\(284.49 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The efficient and effective generation of test-data from high-level models is of crucial importance in advanced modern software engineering. Empirical studies have shown that mutation testing is highly effective. This paper describes how search-based automatic test-data generation methods can be used to find mutation adequate test-sets for *Matlab/Simulink* models.

Keywords: Matlab/Simulink, automation, heuristic search, mutation testing, simulated annealing, test-data generation

3 [Workshop papers: Experiences with model and autocode reviews in model-based software development](#)

Ingo Stürmer, Mirko Conrad, Ines Fey, Heiko Dörr

May 2006 **Proceedings of the 2006 international workshop on Software engineering for automotive systems SEAS '06**

Publisher: ACM Press

Full text available: [pdf\(141.35 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Through the introduction of the model-based development, paradigm models became first class citizens in the development of in-vehicle software and are thus also object to strict quality assurance. Just as code reviews are widespread in classical software development, models also have to undergo a stringent review procedure - particularly if they serve as a basis for automatic software implementation by means of model-based code generators. In addition to model reviews, the generated production c ...

Keywords: autocode review, automatic code generation, complexity, model review, model-based development, modeling guidelines, simulink, stateflow, testing

4 The Role of Model-Level Transactors and UML in Functional Prototyping of Systems-on-Chip: A Software-Radio Application

Alexandre Chureau, Yvon Savaria, El Mostapha Aboulhamid

March 2005 **Proceedings of the conference on Design, Automation and Test in Europe - Volume 2 DATE '05**

Publisher: IEEE Computer Society

Full text available:  pdf(173.08 KB) Additional Information: [full citation](#), [abstract](#), [index terms](#)

Developing a functional prototype of a system-on-chip provides a unifying vehicle for model validation and system refinement. Keeping the prototype executable across several abstraction levels, clock domains and design tools is a key requirement to effective prototyping. This paper presents how model-level transactors address design heterogeneity by unifying event-based and cycle-based worlds from specification to implementation. Transactors are used to build a functional prototype of a software ...

5 Correct-by-Construction Transformations across Design Environments for Model-Based Embedded Software Development

M. Baleani, A. Ferrari, L. Mangeruca, A. L. Sangiovanni-Vincentelli, U. Freund, E. Schlenker, H.-J. Wolff


March 2005 **Proceedings of the conference on Design, Automation and Test in Europe - Volume 2 DATE '05**

Publisher: IEEE Computer Society

Full text available:  pdf(471.02 KB) Additional Information: [full citation](#), [abstract](#), [index terms](#)

Embedded software design for real time reactive system has become the bottleneck in the market introduction of complex products such as automobiles, airplanes, and industrial control plants. In particular, functional correctness and reactive performance are increasingly difficult to verify. The advent of model-based design methodologies has alleviated some of the verification-related problems by making the code-generation process flow automatically from the model description. Given the relative ...

6 Translating discrete-time simulink to lustre

 Stavros Tripakis, Christos Sofronis, Paul Caspi, Adrian Curic

November 2005 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 4 Issue 4

Publisher: ACM Press

Full text available:  pdf(827.48 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We present a method of translating discrete-time Simulink models to Lustre programs. Our method consists of three steps: type inference, clock inference, and hierarchical bottom-up translation. In the process, we explain and formalize the typing and timing mechanisms of Simulink. The method has been implemented in a prototype tool called S2L, which has been used in the context of a European research project to translate two automotive controller models provided by Audi.

Keywords: Code generation, Lustre, Simulink, embedded software

7 Software Engineering for Automotive Systems (SEAS): Towards verified automotive software

 J. Botaschanjan, L. Kof, C. Kühnel, M. Spichkova

May 2005 **ACM SIGSOFT Software Engineering Notes , Proceedings of the second international workshop on Software engineering for automotive systems SEAS '05**, Volume 30 Issue 4


Publisher: ACM Press

Full text available:  pdf(146.34 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Automotive software is one of the most challenging fields of software engineering: it must meet real time requirements, is safety critical and distributed over multiple processors. With the increasing complexity of automotive software, as for example in the case of drive-by-wire, automated driving and driver assistants, software correctness becomes more and more a crucial issue. In order that these innovations can become reality, it is necessary to be able to *guarantee* software correctness ...

Keywords: automotive, integration, model based software engineering, theorem proving, time-triggered, verification

8 Formal languages: Defining and translating a "safe" subset of simulink/stateflow into lustre

 N. Scaife, C. Sofronis, P. Caspi, S. Tripakis, F. Maraninchi

September 2004 **Proceedings of the 4th ACM international conference on Embedded software EMSOFT '04**

Publisher: ACM Press

Full text available:  [pdf\(228.81 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The Simulink/Stateflow toolset is an integrated suite enabling model-based design and has become popular in the automotive and aeronautics industries. We have previously developed a translator called Simtolus from Simulink to the synchronous language Lustre and we build upon that work by encompassing Stateflow as well. Stateflow is problematical for synchronous languages because of its unbounded behaviour so we propose analysis techniques to define a subset of Stateflow for which we can define a ...

Keywords: automatic translation, embedded software, lustre, simulink


9 From simulink to SCADE/lustre to TTA: a layered approach for distributed embedded applications



Paul Caspi, Adrian Curic, Aude Maignan, Christos Sofronis, Stavros Tripakis, Peter Niebert

June 2003 **ACM SIGPLAN Notices , Proceedings of the 2003 ACM SIGPLAN conference on Language, compiler, and tool for embedded systems LCTES '03**, Volume 38 Issue 7

Publisher: ACM Press

Full text available:  [pdf\(243.55 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We present a layered end-to-end approach for the design and implementation of embedded software on a distributed platform. The approach comprises a high-level modeling and simulation layer (Simulink), a middle-level programming and validation layer (SCADE/Lustre) and a low-level execution layer (TTA). We provide algorithms and tools to pass from one layer to the next. First, a translator from Simulink to Lustre. Second, a set of real-time and code-distribution extensions to Lustre. Third, implem ...


Keywords: code distribution, embedded software, lustre, scheduling, simulink, synchronous languages

10 Developing safety-critical systems: the role of formal methods and tools

Constance Heitmeyer

April 2006 **Proceedings of the 10th Australian workshop on Safety critical systems and software - Volume 55 SCS '05**

Publisher: Australian Computer Society, Inc.

Full text available:  [pdf\(184.53 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In recent years, many formal methods have been proposed to improve the quality of safety-critical software systems. These methods include new specification and modeling languages as well as formal verification techniques, such as model checking and theorem proving. This paper describes numerous ways in which tools supporting formal methods can improve the quality of both software code as well as software specifications and models. However, while promising, formal methods and their support tools ...

Keywords: formal methods, formal specification, formal verification, model checking, software testing, software tools, theorem proving


11 Software engineering: applications, practices and tools (SE): High-integrity extreme programming



Richard F. Paige, Howard Chivers, John A. McDermid, Zoë R. Stephenson

March 2005 **Proceedings of the 2005 ACM symposium on Applied computing SAC '05**

Publisher: ACM Press

Full text available:  [pdf\(101.10 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We assess the applicability of Extreme Programming practices to engineering high-integrity systems, focusing on the characteristics of this problem domain that distinguish it from those considered more traditional for agile development. We suggest that Extreme Programming needs both extension and modification to be applicable to engineering high-integrity systems, and discuss promising extensions.

Keywords: agile development, extreme programming, high-integrity systems


12 Specification and synthesis of real-time embedded distributed and parallel multiprocessor-based signal processing systems



Randall S. Janka, Linda M. Willis

November 2000 **Proceedings of the 2000 international conference on Compilers, architecture, and synthesis for embedded systems CASES '00**


Publisher: ACM Press

Full text available:  [pdf\(185.59 KB\)](#)

Additional Information: [full citation](#), [citations](#)

Keywords: COTS, MAGIC, MPI, MPI/RT, VCC, VSIPL, embedded, middleware, multiprocessing, parallel processing, real-time, specification and design methodology, virtual component co-design

13 Echo: a practical approach to formal verification

 Elisabeth A. Strunk, Xiang Yin, John C. Knight

September 2005 **Proceedings of the 10th international workshop on Formal methods for industrial critical systems FMICS '05**

Publisher: ACM Press


Full text available:  [pdf\(270.63 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Safe operation is crucial to safety-critical systems, and formal verification of implementations is a desirable means to increase confidence in safety. Traditional formal verification approaches follow the Floyd-Hoare style, setting up a direct compliance argument between an abstract formal specification and a concrete implementation. Such approaches require proofs of large numbers of verification conditions. Creation of both the conditions and their proofs can be difficult and time-consuming. In ...

Keywords: formal specification, formal verification


14 Guidelines for a graduate curriculum on embedded software and systems

 P. Caspi, A. Sangiovanni-Vincentelli, L. Almeida, A. Benveniste, B. Bouyssounouse, G. Buttazzo, I. Crnkovic, W. Damm, J. Engblom, G. Folher, M. Garcia-Valls, H. Kopetz, Y. Lakhnech, F.

Laroussinie, L. Lavagno, G. Lipari, F. Maraninchi, Ph. Peti, J. de la Puente, N. Scaife, J. Sifakis, R. de Simone, M. Torngren, P. Verissimo, A. J. Wellings, R. Wilhelm, T. Willemse, W. Yi

August 2005 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 4 Issue 3

Publisher: ACM Press

Full text available:  [pdf\(143.68 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

The design of embedded real-time systems requires skills from multiple specific disciplines, including, but not limited to, control, computer science, and electronics. This often involves experts from differing backgrounds, who do not recognize that they address similar, if not identical, issues from complementary angles. Design methodologies are lacking in rigor and discipline so that demonstrating correctness of an embedded design, if at all possible, is a very expensive proposition that may d ...

Keywords: Graduate curriculum, architecture and design, control, distributed systems, embedded systems, extrafunctional properties, labs, real-time

15 SoftContract: an Assertion-Based Software Development Process that Enables Design-by-Contract

Jean-Yves Brunel, Marco Di Natale, Alberto Ferrari, Paolo Giusto, Luciano Lavagno

February 2004 **Proceedings of the conference on Design, automation and test in Europe - Volume 1 DATE '04**


Publisher: IEEE Computer Society

Full text available:  [pdf\(133.58 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [index terms](#)

This paper discusses a model-based design flow for requirements in distributed embedded software development. Such requirements are specified using a language similar to Linear Temporal Logic which allows one to reason about time and sequencing. They consist of assertions which must hold for a design, given some assumptions on its environment. They can be checked both during simulation and, at least for a subset, even on the target. The key contribution of the paper is the extension to the embedd ...

16 An overview of embedded system design education at berkeley

 Alberto L. Sangiovanni-Vincentelli, Alessandro Pinto

August 2005 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 4 Issue 3

Publisher: ACM Press

Full text available:  [pdf\(225.54 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Embedded systems have been a traditional area of strength in the research agenda of the University of California at Berkeley. In parallel to this effort, a pattern of graduate and undergraduate classes has emerged that is the result of a distillation process of the research results. In this paper, we present the considerations that are driving our curriculum development and we review our undergraduate and graduate program. In particular, we describe in detail a graduate class (EECS249: Design of ...

Keywords: Graduate and undergraduate education, architectural design, embedded software, embedded systems, functional design, sourcework

17 Caches and Memory Systems: A vision for embedded software

Alberto Sangiovanni-Vincentelli, Grant Martin

November 2001

Proceedings of the 2001 international conference on Compilers, architecture, and synthesis for embedded systems CASES '01

Publisher: ACM Press

Full text available:  pdf(199.62 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In this paper we describe a vision for the future evolution of Embedded SW (ESW) design methodologies as part of overall Embedded Systems (ES) development. Fundamentally, we believe that the way in which embedded SW is developed today must change radically. The key steps are: first, to link embedded software upwards in the abstraction layers to system functionality; and second, to link embedded software to the programmable platforms that support it. This will provide the much-needed means to ver ...


18 Model based design and test: MATLAB/Simulink for automotive systems design

Jon Friedman

March 2006

Proceedings of the conference on Design, automation and test in Europe: Proceedings DATE '06

Publisher: European Design and Automation Association

Full text available:  pdf(54.81 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

Automotive systems are becoming increasingly difficult and expensive to design successfully as the market demands increasing complexity. Body electronics are particularly affected by this trend, a good example being power windows design. This seemingly mundane area involves meeting market and legislative requirements, which means creating a control system that combines the input from several sensors and follows complex behavioral rules [1].

19 Special session: MATLAB TM - the other emerging system-design language: Matlab extensions for the development, testing and verification of real-time DSP software

David P. Magee

June 2005

Proceedings of the 42nd annual conference on Design automation DAC '05

Publisher: ACM Press

Full text available:  pdf(184.59 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The purpose of this paper is to present the required tools for the development, testing and verification of DSP software in Matlab. The paper motivates a DSP Simulator concept that can be combined with the MATLAB executable interface to develop, evaluate and test DSP software within a single environment. Programming guidelines and optimization results are also provided to demonstrate the effectiveness of the intrinsics software development approach.

Keywords: C intrinsics, DSP software, matlab, optimization, verification

20 W2-C: general symposium: Improved channel codec implementation and performance analysis of OFDM based DAB systems

Yu-Pin Chang, Kai-Sheng Yang, Chao-Tang Yu

July 2006

Proceeding of the 2006 international conference on Communications and mobile computing IWCMC '06

Publisher: ACM Press

Full text available:  pdf(357.90 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

A novel methodology is proposed in this paper to design and implement an improved Digital Audio Broadcasting (DAB) channel codec that is based on Eureka-147 specifications including energy-dispersal, interleaving, concatenated Reed-Solomon and punctured convolutional coding techniques. Furthermore, by utilizing MATLAB/Simulink software, Xilinx System Generator, Xilinx Alliance tools, Xilinx XSE & ISE and Modelsim XE software, a transceiver platform is built-up to simulate and analyze the perfor ...

Keywords: DAB, Eureka-147, OFDM, fading channel, multi-path